Terminology

Active Directory Domain Services - AD DS

Active Directory uses the anology of a forest to explain the structure of an Active Directory.
So you have: Forest, Tree, Branch, Leaf

Forest The forest is the absolut top level of your AD structure. The forest means: all the domains. A forest can contain multiple domain container objects. Ie, many domains. In a company you can a have multiple forests.

Tree A tree in AD referes to a Domain.
The first domain in the forest is called the forest root domain.

Leaf A leaf object is any objects that cannot contain another object. Examples of these are: Users, Computers, Groups.

Design Principles of an Active Directory

An AD can be organized in many different ways, and there is not right answer, but there are many wrong answers. But a good idea is to structure the AD in the same way as the organisation is structured. These principles aren't mutually exclusive, usually they are combined. OUs are flexible objects, and the structure can be changed fairly easily.

- Map the organizations structure - OUs for major business units (HR, Economy, Sales, etc)
- Map the geographical structure - Create OUs based on geography, for example OU for each country, with sub-OUs in each city.
- Create separate OU for workstations and servers
- Create separate OUs for workstations and user - makes it easier to apply relevant GPOs.

Good ideas: - Use a Standardize Naming Convention For example letter of first name + lastname. or "svc_name" for service accounts.

- Don't nest to many levels of OUs. How many is to many? If each OU has a separate GPO, it might taker longer time for the user to log in, because to many GPOs will be applied.

Organizational Unit (OU)

OUs are containers of other AD objects.

OUs are often used to logically separate users into managable units. For example, to have OU for each country where the company has offices. That country OU might then be separated into city specifc OUs. You can thus have an OU inside an OU.
A user cannot belong to two OUs at the same time.
An OU is a container. It is a container because an OU can contain other elements, such as users or computer. A group is not a container. A group is just a simple entry with the attribute "members". The value of that attribute defines who is a member of the group. The group can therefore not contain other entries/objects, like a OU can.

There are basically two reasons to use OU: - Control the delegation of administrative responsibility. So that one user is able to administer all the users in one OU, but no other users. - To apply GPOs on a limited number of users the GPO can be associated with a specific OU.

The OU is the smallest unit to which you can assign a GPO. You can't, for example assign a GPO to a generic container (such as Users and Computers). Those containers inherit the root GPOs though.

It is best practice to not mix computer objects with user objects in OUs.

Managed by

This might seem as an intriguing attribute that all OUs have. Unfortunatly, it is not that interesting as it sounds. It doesn't give the user any additional permissions over the OU. It is more a label to know who is the business owner of that group. So maybe the boss over that business area.

Containers

There are a few differences between a OU and a Container. The main difference is that you can't apply GPOs on containers, only on OUs. A container cannot be created using Active Directory Administrative Center (ADAC) or Active Directory Users and Computers (ADUC). Instead you need to use ADSI Edit, and you user must belong to the Schema Admins security group. By default the only user belonging to that group is the AD account Administrator.

There are some containers that is good to know: - Builtin - stores a number of the default groups - Users - stores other default groups (such as Domain Admins). This is where new users are create if no OU is specified. - Computers - stores computer accounts when a new accounts is created, for example when a computer is joined to the domain.

Active Directory Groups

Groups are used to collect user accounts, computer accounts, and other groups into manageable units. Working with groups instead of with individual users helps simplify network maintenance and administration.

There are two types of groups in Active Directory:

- Distribution groups - Used to create email distribution lists.
- Security groups - Used to assign permissions to shared resources.

For a security perspective it is of course the Security Groups that are of interest to use.

Security Principal

You hear this term quite a bit. In Java and Microsoft terminology a `Security Principal` is an entity that can be authenticated. In practice this means user account, service account, computer account, or thread or process that runs in of a user account or computer account. Each security principal is represented by a unique Security Identifier (SID).

Local Security Principals are managed by the Security Account Manager (SAM).

Security identifiers (SID)

A SID is a value that is used to identify a unique Security Principal. Every time a user signs in the an access token is created for that Security Principal. If you come from a Web background you can simply understand the token as a cookie or a JWT. The token contains the user's SID, user rights, and the SIDs for the groups that the user belongs to.

Apart from these specific SIDs there are also some well known SIDs, like `Everyone` and `World`.

Access token

An access token is a protected object that contains information about a users SID, suer rights and SIDs for groups the user belongs to.

When a user authenticates to a computer the whole authentication is performed. If the authentication is successful the process returns a SID and list of SIDs for the users security groups. The `Local Security Authority` (LSA) uses those SIDs to create an access token.

After LSA create the primary access token a copy of that access token is attached to every process that the user initiates. Whenever an action needs to be performed that requires user rights the operating system need to check that the user has those priviliges. So it checks the associated token to see if it has the correct level of privilige.

So it is just like a web-application.

There are two types of access tokens: `primary` and `impersonating`.

ACL - Access Control List

So when we talk about the possibility to edit an ACL, it is the ACL of a AD object (user, group, computer), not specific files. Access priviliges to certain AD resrouces are configured through Access Control Entries (ACE). An ACL is the collection of Access control entries for a specific object. So each object, such as a computer has an Access Control Entry, which is included in the Access Control List.

So it is basically just a list of who gets access to the object, and what type of access (read, write,modify, delete, etc).

An ACE refers to a specific security principal (User, Group, Computer,e tc).

Example of interesting permissions a user can have of another object is: owner, writeOwner, genericALl, writeDACL.

All these permissions can be abused to escalate priviliges.

DACL/

Discretionary Access Control Lists (DACL). DACL is just the group of ACEs.